

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

In re Application

Inventors: David L. Multer, et al.

Application No.: 09/641,028

Filed: August 17, 2000

Title: BASE ROLLING ENGINE FOR DATA
TRANSFER AND SYNCHRONIZATION
SYSTEM

) PATENT APPLICATION

JUL 24 2003

) Art Unit: 2175

) Examiner: Mofiz, Apu M.

) Customer No. 28554

Technology Center 2100

DECLARATION OF RICHARD M. ONYON
PURSUANT TO 37 C.F.R. §1.131

I, RICHARD M. ONYON, declare that:

1. I am an inventor of the invention described and claimed in the above-identified patent application. I am currently the Chief Executive Officer of fusionOne, Inc., assignee of the instant application, and am a co-founder of the company. I have reviewed the pending application as stated in my Declaration for Patent Application and the pending claims as set forth in the RESPONSE A TO OFFICE ACTION ("RESPONSE A") accompanying this DECLARATION. I have also reviewed U.S. Patent No. 6,295,541 having a filing date of June 7, 1999, and claiming priority to United States Provisional Application Ser. No. 60/069,731, filed Dec. 16, 1997, Ser. No. 60/094,972, filed Jul. 31, 1998, and Ser. No. 60/094,824, filed Jul. 31, 1998.

2. I understand that this Declaration will be filed in the United States Patent and Trademark Office in order to provide factual evidence showing that the invention claimed in the present application was completed prior to the date of December 16, 1997.

3. The facts set forth hereinafter to establish that the claimed invention was completed prior to December 16, 1997 all relate to acts which occurred and were carried out within the United States.

4. In January of 1997, I began working for a company called Cachelink Corporation in Westboro, Massachusetts. Cachelink later changed its name to Mango Software. When I began working at Mango Software, the company was focused on developing a system for "pooling" the disk and memory resources of computers across a TCP/IP network. The product under development embodying this concept was called Medley97. Medley97 was a storage-oriented, distributed network operating system which pooled the resources of network-attached machines to create a single, virtual server.

5. I was hired at Mango Software as the Vice President of Sales and Marketing. My job was to transform Mango Software from an engineering only company into an operating company with sales and marketing functions. I was also charged with taking the Medley97 product to market.

6. Before the launch of Medley 97, and well prior to December of 1997, I had a discussion with one of my former technical employees, Leighton Ridgard, about the technical merits of Medley97. Our discussions included how one could implement a system for keeping files current across a number of machines which were not concurrently connected. Essentially, what I had envisioned was

a synchronization solution, which I personally needed based upon my unrelenting travel schedule and the fact that I owned several computers. With the sync system I envisioned, if my home PC was always in sync with my office PC's files, I would not have to transport my work home at night. Or, if I was going on a business trip, I could simply take my laptop computer without concern for whether or not its contents were completely current. For example, even if I was in Japan, I could plug into the Internet and trust that my laptop's content would soon match my office PC's. While Mango focused on concurrent file access for groups of people across the Internet, I believed there was a bigger opportunity in keeping files "in sync" across multiple computers for one person.

7. The core concept conceived of during the conversation between myself and Mr. Ridgard described above, and which evolved over the months that followed, was originally focused on keeping data files, such as word processing, presentation and spreadsheet files, in sync. Syncing data files is a function which even today is not performed by a number of sync solutions. In general, I needed a system to keeping files "concurrent" across a series of computers which were not located physically near each other in a way that was Internet centric (using, for example, TCP/IP), very efficient (so that even low bandwidth connections could be used to sync a PC) and asynchronous (since all PCs were never likely to be connected at the same time). At that time, all sync products of which I was aware were local in nature, and reconciled differences by comparing information on the two devices to be synchronized at the same time. Such a comparison requires a relatively high speed data connection, such as a physical cable between the devices. Our concept envisioned that a large file belonging to one user could be maintained in synchronization among two or more PC's on the opposite ends of the country through a connection to the Internet because only a tiny "piece" of the file actually needed to be moved across the Internet when a change was detected.

8. In the months following the initial conception of my file sync idea, and prior to December, 1997, I began discussing the file sync concept with Sally Winship (now Sally Winship-Comollo, Vice President of Corporate Communications for fusionOne, Inc.), who was at that time Mango's press relations manager. During these discussions, Ms. Winship-Comollo and I also began discussing my desire to build a new company again. We agreed that the file sync product was a good idea and worthy of pursuit in a new business venture. My discussions with Ms. Winship-Comollo and Mr. Ridgard continued into the fall and winter of 1997/1998 through construction of the prototypes, and intensified as technical issues with Medly97 led to turmoil at Mango Software.

9. Prior to December 1997 Leighton Ridgard and I discussed how to synchronize Outlook data between two PCs. Initially we considered syncing Outlook's entire PST from one place to another, following the idea of syncing a data file. While this would have worked it did not takes us long to realize that Outlook's PST was large and sometimes only a small thing changed such as marking an email as read. Leighton knew of the existence of technologies to compute and efficiently generate the binary difference between two files. Xdelta was one such technology. We quickly decided that we would use binary differencing technology to generate a small difference between the PST files and transmit that difference during the sync. We began investigating acquiring binary differencing technology through purchase or writing it ourselves.

10. It did not take us long to further conclude that syncing PST with binary differences would only work when one wanted both machines to be identical. We quickly realized that in order to have one machine contain a subset of the data found on another machine we would have to parse the PSTs

and extract only the required data. We decided at this point to use Binary differencing would work for files. Outlook and other PIM content would need a different methodology.

11. Well prior to December 1997, I had several meetings with Ms. Winship-Comollo where we flushed out further details of the file sync system design. During at least one of these meetings, using the white board in her office, I drew block diagrams of the "store and forward" system. According to my original conception of a store and forward system, when a change is made to file data on a first computer, those changes would be recorded and transported via a network to be stored on an FTP server. At some later point in time, whether or not the first computer was coupled to the network, those changes could be forwarded to a second computer. In this conception, I envisioned some triggering event to forward and retrieve changes to and from the sync server. These drawings were erased from her whiteboard at the conclusion of our meeting and no copies of these drawings were made.

12. During our discussions regarding transmitting only that piece of the file which had changed, we realized that the method of creating and recording changes to only that portion of the data which had changed meant we needed some representation of the data file prior to the change. This need for a known state of a data file evolved into the so-called application object store or "AOS". The AOS is essentially a known state of a user's data as of the last time our system checks to determine whether any changes are made to the user's data. The construction of the prototype of the AOS application began prior to December of 1997, with numerous substantive technical discussions regarding the application taking place prior to October 1997.

13. It was at this point, again well prior to December 1997, that we invented our change log and AOS technology. It was decided that we would need to extract from the PIM each item individually and transmit that item efficiently across to the other device. In order to do this we would need to compute the difference between the PIM data now and the PIM data at the time of last sync. Having already spent time examining delta technology (which we decided we would use for unstructured data such as files) we then created what we called "Structured Delta". Two systems of discovering the "difference" between the PIM now and when we last synced it was created. One was to monitor the PIM for user activity (changing, records, adding records etc.). Another was to store a copy of the PIM data at the end of the sync (in our AOS) and then to compare the stored copy with the PIM copy and determine the differences. Regardless of the method used to compute the differences, we created a system of representing the differences by using a set of operations followed by the data used by that operation. In other words, we created a log of activity (also called a "change log" or "data package") that contained instructions (such as add, delete, modify, rename, and move) along with the relevant data.

14. When creating the change log, we realized we should provide information in a header for the change log to describe the contents of the change log. At this point in time it was decided that the header should contain a signature, the identity of the generating device, the date, the PIM used, version schema in use etc.

15. Still well prior to December 1997, we realized that someone who syncs frequently would generate a lot of these change logs. At this time, we thought they would be stored on an FTP site, and storage space was a consideration. We further discovered that even after items were deleted their

previous existence would be evident in the old change logs. In order to more efficiently manage the users space it was decided that a process could be developed to combine the change logs. We called this process "log rolling" at the time. Initially we thought about running this process on a back-end server without even involving the user. Such a process would apply successive change logs to each other and combine changes and deletes etc. to produce a minimal set of resultant changes collapsed into ideally one change log. During these discussions we decided that even if not done behind the scenes on the server, this feature would be equally valuable if implemented on the client. We later decided to called server-side log rolling "Base rolling" and left the term "log rolling" for use in describing client-side rolling.

16. Our discussions on finalizing the structure proceeded rapidly and within two weeks of when we started we had a thorough understanding of what we wanted to build.

17. After this initial series of technical discussions, I asked Mr. Ridgard to construct a prototype of the system based on these discussions. I recall this construction beginning prior to December of 1997 and being completed in or about February 1998. I am aware that his recollection is that he began later. Despite my eagerness to get started, Leighton could not begin until then because he was employed elsewhere and had other responsibilities. In the interest of time (and because he was the only person working on this) it was decided to produce a proof of concept prototype which synced the entire PST from one device to the other. Although this was certainly not the implementation upon which we had decided, it was sufficient as a proof of concept. In the spring of 1998 two other engineers joined him and we produced the second prototype which implemented an early version of our change log technology.

18. Throughout our development of our system, we worked diligently to reduce the invention to practice. This included the creation of at least three (3) prototypes throughout the process of hiring engineers and gaining funding to complete development of the system.

19. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date:

7/16/03

By:



Richard M. Onyon



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED



re Application

Inventors: David L. Multer, et al.

Application No.: 09/641,028

Filed: August 17, 2000

Title: BASE ROLLING ENGINE FOR DATA
TRANSFER AND SYNCHRONIZATION
SYSTEM

) PATENT APPLICATION

JUL 24 2003

) Art Unit: 2175

Technology Center 2100

) Examiner: Mofiz, Apu M.

) Customer No. 28554

DECLARATION OF LEIGHTON RIDGARD
PURSUANT TO 37 C.F.R. §1.131

I, LEIGHTON RIDGARD, declare that:

1. I am an inventor of the invention described and claimed in the above-identified patent application. I am currently an Principal Software Engineer of fusionOne, Inc., assignee of the instant application. I have reviewed the pending application as stated in my Declaration for Patent Application and the pending claims as set forth in the RESPONSE A TO OFFICE ACTION ("RESPONSE A") accompanying this DECLARATION. I have also reviewed U.S. Patent No. 6,295,541 having a filing date of June 7, 1999, and claiming priority to United States Provisional Application Ser. No. 60/069,731, filed Dec. 16, 1997, Ser. No. 60/094,972, filed Jul. 31, 1998, and Ser. No. 60/094,824, filed Jul. 31, 1998.

2. I understand that this Declaration will be filed in the United States Patent and Trademark Office in order to provide factual evidence showing that the invention claimed in the present application was completed prior to the date of December 16, 1997.

3. The facts set forth hereinafter to establish that the claimed invention was completed prior to December 16, 1997 all relate to acts which occurred and were carried out within the United States.

4. I have been personally associated with Rick Onyon since November of 1991. I met Rick in November 1991 when he was looking for an engineer to work on an idea he had for hierarchical storage for PCs. I worked on that product as one of two engineers while I was employed at KnowledgeWare Inc. In February 1992 I left KnowledgeWare to work for Rick's new company Chili Pepper Software. Chili Pepper software was purchased by Cheyenne Software which was later purchased by Computer Associates. I worked with Rick until he left Computer Associates to go to Mango Software in early 1997. Although at this point in time we were working for separate companies, we remained in touch and often spoke about his ideas for starting a new company and a new product to do file synchronization. In February of 1998, I was hired by Mango Software and once again was working with Rick. Rick eventually quit Mango Software, moved to California and started fusionOne. Once funding was in place I quit Mango Software and went to work for Rick in November of 1998 at fusionOne.

5. During all of that time I was living in Atlanta. Rick had moved to New York when Cheyenne acquired Chili Pepper. He would later move to Boston when he went to work for Mango Software. We remained in touch often. Whenever business or personal travel brought him to or through Atlanta

we would get together socially. These meetings almost always involved discussing Rick's latest product ideas.

6. It was at one of these meetings well prior to December of 1997 that Rick introduced me to his latest idea for a file synchronization service. His initial idea was to create a virtual drive on each machine called the "I" drive. The basic idea was that anything you place in that virtual drive would be kept in sync with all other machines also using the "I" drive. Rick would later decide to call this product "OneDrive", had decided on a company name and had registered domain names for this name. None of these product or domain names were used because by then the product had evolved into so much more.

7. Admittedly I was not initially impressed with the OneDrive idea. Nevertheless, we continued to have conversations about how to sell it, how to use FTP storage, partnering with ISPs, and other aspects of the system.

8. Within a few weeks of our initial conversation on this idea, and still well prior to December 1997, Rick realized that he wanted to synchronize more than just his files. He also wanted his emails, calendar items, and contacts to be kept in sync between his home and work PCs. I offered that we could simply keep the outlook PST in sync with the same techniques we were planning to use to keep the I-Drive files in sync. The outlook PST tends to be much larger than the typical files we had expected to keep in sync. This quickly led us to the conclusion that our sync engine needed to be efficient about how much data it transmitted between PCs to keep them in sync. I speculated that we could use some sort of binary differencing methodology to generate small delta files between a

current PST and a historical PST. Such technology was available and we would either incorporate an existing one or write our own.

9. It became apparent to us that the binary differencing idea did not make any sense when applied to a PST. If we used this method then we would be producing an exact duplicate of the Outlook PST on the second machine. This was a problem because: (1) a user may not want their personal family information stored on their home PC to be synced to their work PC; (2) sensitive office data in emails should not be synced to the home PC, and (3) we would not be able to support a situation where the two PCs were not running the same version of Outlook. We also realized that a huge beneficial side-effect of creating a system extracting specific information from Outlook was that would be able to extend it to keep two totally different personal information managers or "PIMs" in sync. At this point we decided that binary differencing would only be used for generic files but not for PIM data.

10. We needed to create a technology that would allow us to extract the specific information in one PIM that the user specified, encode it efficiently, transfer it to another machine, and then apply it to the destination PIM. Outlook was our primary focus, but we were mindful of other PIMs while we pursued this goal. Rick established the guideline that the solution needs to be able to sync data in Outlook on one PC into an empty copy of Outlook on another PC with no loss of data. Furthermore successive syncs should be accomplished with the minimal amount of data transfer.

11. I thought about this a lot and eventually with Rick came up with the model of recording transactions on the source machine which would be played back on the target machine. In keeping

with the original file sync idea we kept in place the idea of using a temporary holding place (such as FTP storage at one's ISP) for these transaction logs. This would eventually be called store-and-forward sync.

12. From this point forward, and still prior to December 1997, we had a clear direction of every next step to be followed.

13. Next, and still prior to December 1997, we went about devising a quick way to generate the transactions. Simply looking at a PST would not tell us what was different since the last time we synced. We quickly hit upon two ideas. One was to monitor the PIM and record the user activity in the transaction log. The other was to store a historical copy of the PIM data and then compare the current PIM data with the historical copy and generate transactions from the difference. It was quickly decided that we would do both. We would use monitoring for PIMs that could support it and we would use historical comparison for all other PIMs.

14. Mostly because of Rick's insisting that we support every field in Outlook and all other PIMs (current and future), I was driven to device a database scheme that could be used to store the historical data. This would eventually be known in our product as the AOS (Application Object Store). I concluded that for example: a contact from Outlook with a certain set of fields and a contact from Organizer which may contain a different set of fields needed to be stored in the same database. This led to the design of creating a database that was a storage facility for 'fields' instead of the usual database methodology of storing 'records'. The result was a database that could contain the superset of all PIM data across all applications current and future.

15. During the following months all prior to December 1997, Rick and I would continue to develop these ideas over the phone and on his occasional visits to Atlanta. This included devising a procedure for generating the transaction logs from the difference between the current PIM data and the data stored in the AOS. We also wanted to ensure that files could still be kept in sync using the same or very similar technique. We came up with the idea that we needed a differencing engine that could compute the differences of not only PIM data but binary files as well. This differencing engine would eventually be known as Structured Delta. The key design feature we came up with at the time was that the differencing engine would not actually perform a sync per se. Instead the differencing engine had a dual role. One was to determine what changed, generate transaction logs (later to be called change logs), and transmit them to the central store (FTP site). The other role was the reverse. It would get the logs from the central store, decode the logs, and apply the transactions to the target PIM. Sync was actually accomplished by the sequential use of the differencing engine on the source and then target machines.

16. The final task was to design the transaction logs themselves. This was fairly straight forward given our database design of storing items at the field level. The differencing engine would determine what was added, moved, renamed, changed, or deleted and would do this at the field level. It would generate into our change log the list of operations (add, delete, move, etc.) along with the list of fields that were associated with that operation. A header would be placed at the beginning of the file to include items like the schema version, file signature, date generated, PIM used, device used etc. This was done based on our experience as programmers. Schema version would be used so we could revise the format in the future if required. The signature was used so the file integrity could

be validated before use etc. These things are common in most file formats and we decided to do likewise in our new file format.

17. Still well prior to December 1997, we had a concrete idea of the aforementioned aspects of the system. Further discussions would lead us to a couple more important enhancements to our design.

18. The first enhancement was that we could treat file synchronization just like PIM sync. Files were just like all other data in our model because they could be thought of as consisting of a set of fields (name, date, size, body, etc.). However we decided to keep the original idea of using binary differencing for the file body. So we incorporated into Structured Delta the methodology of encoding into Change logs the data generated from the binary difference between the current file and the historical copy.

19. The second enhancement came from mental field tests of the now rapidly evolving sync product design. We played out the scenario of a busy user who syncs a lot. We realized that such a user would have stored in their FTP site many Change logs that contained historical data. For instance if the contact had been deleted the old change logs would still contain historical useless information about that contact. In order to use the user space more responsibly we decided that periodically we could run a process against the central store that would collapse all of the change logs into one. Initially we thought about running this process on the server without even involving the user. Such a process would apply successive change logs to each other and combine changes and deletes etc. to produce a minimal set of resultant changes collapsed into ideally one change log.

During these discussions we decided that even if not done behind the scenes on the server this feature would be equally valuable if implemented on the client. We later decided to call server-side log rolling "Base rolling" and left the term "log rolling" for use in describing client-side rolling. The term "base" was used to denote the concept that this would establish a base starting point for future changes. In other words, that Base log combined with future change logs would comprise all of one's PIM data. When base rolling was done in the future it would then establish a new base.

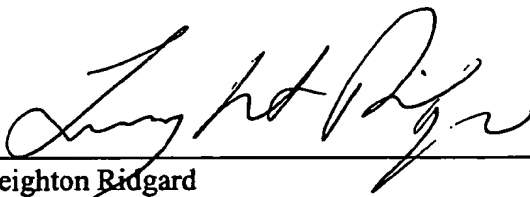
20. Due to my responsibilities at my place of employment, I did not have significant time to work on the next step which was to build a prototype. I recall that at Rick's request I began building a prototype in February 1998. I understand that Rick believes this occurred earlier. In the interest of time, it was decided to produce a proof of concept prototype which synced the entire PST from one device to the other. Although this was certainly not the implementation which we had decided upon, it was sufficient as a proof of concept. In the spring of 1998, Liam Stannard and Rob Garner started working with me to produce the second prototype. The second prototype was completed in the summer of 1998 and implemented an early version of our change log and structured delta design.

21. In the summer of 1998, David Multer, Rob Garner, Liam Stannard, Don Cash, and myself met in Atlanta to produce the official architecture of the product. This occurred rapidly due to the concrete conception we had brought to that meeting. In November of 1998 we started working for fusionOne implementing that design in a commercial product.

23. Throughout our development of our system, we worked diligently to reduce the invention to practice. This included the creation of at least three (3) prototypes throughout the process of hiring engineers and gaining funding to complete development of the system.

24. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: JULY 16TH 2007

By: 
Leighton Ridgard